Oracle Day 5 – Single Row Functions

Note: Please watch my YouTube sessions to better understand the descriptions and queries below

NiC IT Academy YouTube Videos for reference

🔵 Oracle SQL Tutorial - English

https://youtube.com/playlist?list=PLsphD3EpR7F9mmtY2jBt_O8Q9XmvrhQEF

🟡 Oracle SQL - தமிழில்

https://youtube.com/playlist?list=PLsphD3EpR7F-u4Jjp_3fYgLSsKwPPTEH4

📌 Oracle SQL Day wise Video: ENGLISH

Oracle SQL Day 1 – Introduction to Oracle - https://youtu.be/hLnKjYGr730

Oracle SQL Day 2 – SQL Types DDL, DML, DRL, DCL, TCL - https://youtu.be/XpgjXvnfZec

Oracle SQL Day 3 – Constraints in Oracle - https://youtu.be/TmYqeFfHyyc

Oracle SQL Day 4 – SELECT Statements in Oracle - https://youtu.be/tYQfBgUCpoI

Oracle SQL Day 5 – Single Row Functions in Oracle - https://youtu.be/4qJJxQuHLC4

Oracle SQL Day 6 – Joins in Oracle - https://youtu.be/CkaqluC2afE

Oracle SQL Day 7 – Aggregate Functions in Oracle - https://youtu.be/BSiCWzj-py8

Oracle SQL Day 8 – Sub Queries in Oracle - https://youtu.be/KtUCyG2cZe4

Oracle SQL Day 9 – SET Operators in Oracle - https://youtu.be/B0JbGbWsEIA

Oracle SQL Day 10 – Analytical Functions in Oracle - https://youtu.be/gRC3ndWLsoo

Oracle SQL Day 11 - Views in Oracle - https://youtu.be/m8a1UtOmd5k

Oracle SQL Day 12 - Indexes in Oracle - https://youtu.be/reL2O-kvNxc

Oracle SQL Day 13 - Regular Expression - https://youtu.be/k_Eo08vLPhU

Single Row functions in Oracle:

```
===========================

select first_name,upper(first_name),lower(first_name),initcap(first_name),
length(first_name),reverse(first_name) from employees;

select * from employees where first_name='John';

select * from employees where first_name='john';

select * from employees where lower(first_name)='john';

select * from employees where UPPER(first_name)='JOHN';

select address,lower(address) from customer_details;

select address from customer_details where lower(ADDRESS)='chennai';

select * from customer_details where address ='chennai';

select * from customer_details where lower(address) ='chennai';

select * from customer_details where upper(address) ='CHENNAI';

-- Substr  --sub string

substr(string,from_position,no_of_char);    -- 3 argument

substr(string,from_position);    -- 2 argument
```

```
select substr('Welcome to India!!!',12,5) from dual;

select substr('Welcome to India!!!',12) from dual;

select substr('Welcome to India!!!',-8) from dual;

select substr('Welcome to India!!!',-8,5) from dual;

select job_id,substr(job_id,1,4) from employees;

-- Instr will return the position of the character

select INSTR('CORPORATE FLOOR','R') from dual;

select INSTR('CORPORATE FLOOR','AB') from dual;

-- from 3rd position 2nd occurrence
select INSTR('CORPORATE FLOOR','OR',3,2) from dual;

select INSTR('CORPORATE FLOOR','OR',3,1) from dual;

select substr('When system dialog prompts, click Open Zoom Meetings.',1,
instr('When system dialog prompts, click Open Zoom Meetings.',',')-1) from dual;

select substr('sample@gmail.com', 1, instr('sample@gmail.com', '@')-1) from dual;

select substr('abc@gmail.com', 1, instr('abc@gmail.com', '@')-1) from dual;
```

```sql
select INSTR('CORPORATE FLOOR GARDEN',' ') from dual;

select substr('CORPORATE FLOOR GARDEN',17) from dual;

select instr('CORPORATE FLOOR GARDEN',' ',1,2) from dual;

select substr('CORPORATE FLOOR GARDEN',instr('CORPORATE FLOOR GARDEN',' ',1,2)+1) from dual;

select substr('WELCOME TO CHENNAI CHROMPET',instr('WELCOME TO CHENNAI CHROMPET',' ',1,3)+1)
from dual;

select substr('asfdfadsfad@gmail.com', 1, instr('asfdfadsfad@gmail.com','@')-1) from dual;

--extract middle name from full_name

select substr('NIC IT ACADEMY',instr('NIC IT ACADEMY',' ')+1,
instr('NIC IT ACADEMY',' ',1,2)-instr('NIC IT ACADEMY',' ')) middle_name from dual;

select substr(full_name,instr(full_name,' ')+1,
instr(full_name,' ',1,2)-instr(full_name,' ')) middle_name from customer;


select LPAD('WELCOME',15,'*') from dual;

select RPAD('WELCOME',15,'*') from dual;

select LPAD(RPAD('WELCOME',15,'*'),30,'*') from dual;
select salary,LPAD(salary,15,0) from employees;

select LTRIM('        Welcome') from dual;

select RTRIM('        Welcome      ') from dual;

select TRIM('                         Wel come      ') from dual;

select LTRIM('00000000000100123', '0') from dual;

select LTRIM('00101233234345354650121211', '021') from dual;

select RTRIM('00101233234345354650121211', '021') from dual;

select LTRIM(RTRIM('00101233234345354650121211', '021'),'021') from dual;



select REPLACE('JACK and JUE','J','BL') from dual;

select REPLACE(JOB,'MANAGER','BOSS') from EMP;

job:
===
Manager     BOSS
manager     BOSS
MANAGER     BOSS

select REPLACE(upper(JOB),'MANAGER','BOSS') from EMP;
select phone_number,replace(phone_number,'.',null) from employees;
```

515.123.4567   ==> +1-51512-34567

```
select phone_number,'+1-'||substr(replace(phone_number,'.',''),1,5)||'-
'||substr(replace(phone_number,'.',''),6,5)||'-'||
substr(replace(phone_number,'.',''),11,5) new_ph_num from employees;
```

515.123.4567

```
select phone_number,
case
when length(phone_number)=12 then '+1-'||substr(replace(phone_number,'.',''),1,5)||'-
'||substr(replace(phone_number,'.',''),6,5)
else
'+1-'||substr(replace(phone_number,'.',''),1,5)||'-'||substr(replace(phone_number,'.',''),6,5)||'-'||
substr(replace(phone_number,'.',''),11,5) end new_ph_num from employees;
```

translate - position wise translation

ABCD   XYZ

A--> X

B--> Y

C--> Z

D--> NUll

WBADCS --> WYXZS

select translate('WELCOME TO CHENNAI','ABCDEF','WXYZ') from dual;

A	W
B	X
C	Y
D	Z
E	null
F	null

WLYOM TO YHNNWI

--Dealing with Null values:

Any arithmatic operations on null values results null

value*null ==> null

value+null ==> null

value-null ==> null

NVL - 2 arg

NVL2 - 3 arg

Nullif - 2 arg

Coalesce - n arg

NVL(arg1,arg2)

  if arg1 is null ---> arg2

  if arg1 is not null --> arg1

```
select NVL(5,6) from dual;  -- 5
```

```
select NVL(null,6) from dual; --6
```

```
select * from employees;
```

```
select employee_id,salary,commission_pct,salary+(salary*commission_pct) total_salary from employees;
```

```
select employee_id,salary,commission_pct,salary+(salary*nvL(commission_pct,0)) total_salary from employees;
```

NVL2(arg1,arg2,arg3)

  if arg1 is null ---> arg3

  if arg1 is not null --> arg2

```
select nvl2(4,8,12) from dual;  --8

select nvl2(null,8,12) from dual; -- 12

select * from employee;

select employee_id,emp_name,allocation_id,nvl2(allocation_id,'Allocated','Waiting for project')
allocation_status
from employee;
-------------------------------------------------------------
```

nullif(arg1,arg2)

```
  if arg1=arg2   ---> null
  if arg1 != arg2   ---> arg1

select nullif(5,8) from dual; --5

select nullif(8,8) from dual; -- null

select first_name,last_name from employees where first_name=last_name;

select first_name,last_name from employees where nullif(first_name,last_name) is null;

-------------------------------------------------------------
```

coalesce(arg1,arg2,arg3.. ..arg_n)

-- It will return first not null value

COALESCE -- It will always return first not null value

select commission_pct,manager_id,department_id from employees;

select commission_pct,manager_id,department_id,

COALESCE(commission_pct,manager_id,department_id,0) from employees;

select employee_id,salary,commission_pct,salary+(salary*COALESCE(commission_pct,0)) total_salary from employees;

select coalesce(mobile_no,office_no,resi_no,'no_phone_number') from dual;

----

select round(0.7) from dual;

select round(2.8) from dual;

select round(4.35) from dual;

select round(5435.7878) from dual;

select round(5435.3878) from dual;

```sql
select  round(5435.7878,2) from dual;

select  round(5435.9978,2) from dual;

select  round(5435.783258,3) from dual;

select  round(5435.783258,4) from dual;

select  round(5435.7878,-2) from dual;

select  round(5475.7878,-2) from dual;

---------------------------------------------------
-- Trunc will always take base value

select  trunc(0.7878) from dual;

select  trunc(5435.7878) from dual;

select  trunc(5435.3878) from dual;

select  trunc(5435.7878,2) from dual;

select  trunc(5435.783258,3) from dual;

select  trunc(5435.783258,4) from dual;
```

-- it will remove timestamp from a date&time, returns date part alone

select SYSTIMESTAMP from dual;

select trunc(SYSTIMESTAMP) from dual;

----

ceil -- always top value

Floor -- Always base value

select floor(5.99999999) from dual;

select floor(5.000000999) from dual;

select ceil(5.99999999) from dual;

select ceil(5.000000001) from dual;

what is the difference between trunc and floor?

-- mod returns reminder in the division operation

select mod(55,4) from dual;   -- 55/4 reminder 3

select mod(55,3) from dual;   --55/3   reminder 1

-- leap year of hire_date

select * from employees where mod(to_char(hire_date,'yyyy'),4)=0;

```
select * from employees where mod(to_char(hire_date,'yyyy'),4)!=0;   -- not equal   <>  !=
```

```
-- even number of employee_id

select * from employees where mod(employee_id,2)=0;
```

----

months between two dates

```
-- MONTHS_BETWEEN(date1,date2)
```

```
select  MONTHS_BETWEEN (TO_DATE ('2020/01/01', 'yyyy/mm/dd'), TO_DATE ('2010/01/01',
'yyyy/mm/dd') ) total_months from dual;
```

```
select months_between(sysdate, TO_DATE ('2000/01/01', 'yyyy/mm/dd')) from dual;
```

-- days between two dates

```
select date2 - date1 total_days from dual;
```

```
select TO_DATE ('2017/01/01', 'yyyy/mm/dd')- TO_DATE ('2014/01/01', 'yyyy/mm/dd') from dual;
```

-- next weekday of given date

```
select NEXT_DAY('31-Mar-20', 'FRIDAY') from dual;
```

```
select next_day(sysdate,'Thursday') from dual;
```

```
-- last day of given month

select LAST_DAY(sysdate) from dual;

select LAST_DAY(sysdate+25) from dual;

sysdate=last_day(sysdate)

-- add_months(date,number_of_months)

select ADD_MONTHS('01-Aug-03', 3) from dual;

select ADD_MONTHS(sysdate, 60) from dual;
```

--Round and truncate of Dates:

Year

Q

Month

Day

year --> half + half ====> 6 months + 6 months

if any date falls on first half of the 6 month --> first_day of the year

if any date falls on second half of the year (second six months) --> first_day of the next year

```
select ROUND(TO_DATE ('22-AUG-21'),'YEAR') from dual;
```

```
select ROUND(TO_DATE ('22-Apr-21'),'YEAR') from dual;
```

```
select ROUND(sysdate,'YEAR') from dual;
```

```
-- Last day of the year
```

```
select ROUND(sysdate,'YEAR')+360 from dual;
```

```
select last_day(ROUND(TO_DATE ('22-Apr-21'),'YEAR')+360) from dual;
```

Quarter ==> 90 days ==> 45 days + next 45 days

if any date falls on first half of the Quarter --> first_day of the quarter

if any date falls on second half of the Quarter (second 45 days) --> first_day of the next quarter

e.g JAN FEB MAR ==> Jan 1 to Feb 14 --> first half of the Q
Feb 15 to March 31 --> second half of the Q

```
select ROUND(TO_DATE ('22-AUG-16'),'Q') from dual;
-- jul,aug,sep --> 2nd half of Q3 --> first day of Q4
```

```
select ROUND(TO_DATE ('13-Apr-20'),'Q') from dual;
select ROUND(TO_DATE ('02-NOV-20'),'Q') from dual;

select ROUND(TO_DATE ('22-AUG-16'),'MONTH') from dual;

select ROUND(TO_DATE ('13-Apr-16'),'MONTH') from dual;

select ROUND(TO_DATE ('02-NOV-16'),'MONTH') from dual;

-- Weekday  3.5 days

 Sun  Mon  Tue  Wed+    Wed(12hrs)  Thur  Fri Sat       Sun

select ROUND(TO_DATE ('22-AUG-16'),'DAY') from dual;

select ROUND(sysdate+4,'DAY') from dual;

select ROUND(TO_DATE ('30-NOV-16'),'DAY') from dual;

--------------------------------------------------------------------------------

select TRUNC(TO_DATE ('22-AUG-16'),'YEAR') from dual;

select TRUNC(TO_DATE ('22-Apr-16'),'YEAR') from dual;

select TRUNC(TO_DATE ('22-AUG-16'),'Q') from dual;
select TRUNC(TO_DATE ('13-Apr-16'),'Q') from dual;

select TRUNC(TO_DATE ('02-NOV-16'),'Q') from dual;

select TRUNC(TO_DATE ('22-AUG-16'),'MONTH') from dual;

select TRUNC(TO_DATE ('13-Apr-16'),'MONTH') from dual;

select TRUNC(TO_DATE ('02-NOV-16'),'MONTH') from dual;

select TRUNC(TO_DATE ('22-AUG-16'),'DAY') from dual;

select TRUNC(sysdate+5,'DAY') from dual;

select TRUNC(TO_DATE ('30-NOV-16'),'DAY') from dual;

select sysdate+3   from dual;
```

first day of the month:

=======================

```
select to_date(to_char(sysdate,'yyyymm')||'01','yyyymmdd') from dual;

select TRUNC(sysdate,'Month') from dual;

select trunc(LAST_DAY(sysdate),'Month') from dual;

select last_day(ADD_MONTHS(sysdate, -1))+1 from dual;
```

find first day and last day of quarter:

=============================

```
select TRUNC(sysdate,'Q') from dual;

select last_day(TRUNC(sysdate,'Q')+75) from dual;

select last_day(TRUNC(TO_DATE ('13-Apr-16'),'Q')+75) from dual;
```

Find First and Last Day of the last Quarter in ORACLE

===

```
SELECT
  ADD_MONTHS(TRUNC(SYSDATE, 'Q'), -3) AS First,
  TRUNC(SYSDATE, 'Q') - 1 AS Last
FROM DUAL;
```

```
select TO_CHAR(1210.73, '9999.9') from dual;

select TO_CHAR(1210.78, '$9999.9') from dual;

select TO_CHAR(1210.73, '$9,999.999') from dual;

select TO_CHAR(sysdate, 'yyyy/mm/dd') from dual;
select TO_CHAR(sysdate, 'Mon-ddth-yyyy') from dual;

select TO_CHAR(sysdate, 'MM-ddth-yyyy') from dual;

select to_number(TO_CHAR(sysdate, 'mmddyyyy'))+1 from dual;

select to_char(sysdate+1,'mmddyyyy') from dual;

select TO_CHAR(sysdate, 'HH24:MI:SS') from dual;

select TO_CHAR(sysdate, 'mm/dd/yyyy HH24:MI:SS') from dual;

select TO_CHAR(sysdate, 'HH12:MI:SS AM') from dual;

------------------------------------------------------------------------------
select abs(-354) from dual;
------------------------------------------------------------------------------
```

decode

The DECODE function in Oracle allows you to have IF-THEN-ELSE logic in your SQL statements.

The expression is the value to compare. Many combinations of search and result

can be supplied. Search is compared against the expression, and if it is true, then result is returned

City	New_city
Madras	Chennai
Calkatta	Kolkatta
Bombay	Mumbai

Orissa Odisa

any other city -- > city

```
decode(city,'Madras','Chennai','calcatta','Kolkatta','Bombay','Mumbai','Orissa','Odissa',city) new_city
```

```
select subject_id,
decode(subject_id,1,'Mathematics',2,'Physics',3,'Chemistry','Others') subject_name
from students;
```

f

F	Female
M	Male
U	Unknown

```
decode(upper(gender),'F','Female','M','Male','U','Unknown','Others')
```

transaction_status

S	success
F	FAILED
P	pending
A	approved
U	unknown

example:

Transaction_status_code to Transaction_status:

'S' Success

F Failed

P - pending

U unknown

Case:

```
case when condition1 then statement1
    when condition2 then statement2
    when condition3 then statement3
else
    statement
end;
```

salary_status:

----

salary < 5000   Low salary

salary >= 5000 <15000 Avg salary

salary >=15000 high salary

```
select employee_id,first_name,salary,
case
when salary < 5000 then  'Low salary'
when salary >= 5000 and salary <15000 then 'Avg salary'
```

```
else 'High Salary' end salary_status

from employees;


select employee_id,first_name,salary,

case

when salary < 5000 then  'Low salary'

when salary >= 5000 and salary <15000 then 'Avg salary'

else 'High Salary' end salary_status,

case

when salary < 5000 then  salary+(salary*0.3)

when salary >= 5000 and salary <15000 then salary+(salary*0.2)

else salary+(salary*0.1)end new_salary

from employees;



select count(case when salary < 5000 then  'Low salary' end ) as low_salary_count,

count(case when salary >= 5000 and salary <15000 then  'Avg salary' end ) as avg_salary_count,

count(case when salary >= 15000 then  'high salary' end ) as high_salary_count

from employees;


select sum (case when id >= 0 then id end) as positive,

     sum (case when id < 0 then id end) as negative

from customer2;


-- Exercise

-- Find the second Saturday of the given month


SELECT NEXT_DAY(NEXT_DAY((TRUNC(TO_DATE('01-MAR-2019', 'DD-MON-YYYY'),

                         'MONTH') - 1),
                    'SATURDAY'),
               'SATURDAY') SECOND_SATURDAY
  FROM DUAL;
```

```
select (TRUNC(sysdate,'MONTH') - 1) from dual; -- last day of the previous month
```

```
select NEXT_DAY((TRUNC(sysdate,'MONTH') - 1),'SATURDAY') from dual;
```

```
select NEXT_DAY(NEXT_DAY((TRUNC(sysdate,'MONTH') - 1),'SATURDAY'),'SATURDAY') from dual;
```

===